Visualization of Spatial Knowledge with Ontology Trees and Adaptable Search Result Grids in the Era of Web 3.0

Matthias Loskyll, Dominikus Heckmann
(Saarland University/DFKI GmbH, 66123 Saarbrücken, Germany
matthias@xantippe.cs.uni-sb.de)

Christian Glahn
(Open University, Netherlands
christian.glahn@ou.nl)

Abstract: With the emerging trend of Web 3.0 and the resulting huge amount of user-generated semantically-enriched data, improved ways of knowledge visualization and human computer interaction are needed. We present several techniques of visualizing particularly spatial knowledge in largely scalable, clear structured ontology trees on the web. In addition, we describe the representation of search results with a combined approach consisting of Ajax-based grids and Google Maps.

Key Words: spatial knowledge visualization, search results visualization, ontology
Category: H.5.2

1 Introduction

The visualization of spatial knowledge in the era of Web 3.0 raises interesting new research questions. How can one represent huge collections of knowledge (e.g. ontologies with over 10 million concepts) as browsable trees in a scalable manner and with a clear user interface? Which is an easy-to-use and efficient representation of search results, especially over spatial data? These are the central questions we want to address with our visualization approaches.

The visualization techniques we present in this paper are part of the **UbisWorld** project, which focuses its research on ubiquitous user modeling [Heckmann 2006] and Web 3.0, i.e. the integration of the ideas of the Web 2.0 with the technologies of the Semantic Web [Wahlster and Dengel 2006]. The knowledge base of the system is built out of two ontologies: GUMO (General User Modeling Ontology) [Heckman et al. 2007] and UbisOntology. Several external ontologies (e.g. SUMO [Pease 2002], DOLCE, OpenCyc) and taxonomies (e.g. the Amazon.com category tree) have also been parsed and integrated into the UbisWorld.

The ontologies are represented as foldable trees whose nodes consist of classes and instances whereby the conversion from ontologies (usually represented as graphs) to trees is performed by using multiple heritage. All the trees can be browsed in a very efficient way by using Ajax technologies [Loskyll 2007] [Garrett 2005].

This means that the data needed to display the layer of a tree is sent to the client not before the user has opened the corresponding parent node. Using this visualization technique we are able to display huge ontologies as trees. In addition to classes and instances, we are also able to visualize relations and properties as well as limitation nodes which help to make the trees easier to browse. While other approaches like the GeoTree (http://geotree.geonames.org/) only display at most 999 nodes per tree layer, we are basically able to display arbitrary large numbers of nodes on one layer and to keep a clear structured user interface at the same time.
The second methodology described in this paper aims at an easy-to-use representation of search results. Again we use Ajax technologies to create an adaptable search result grid, which provides paging and sorting functionalities. Additionally, the currently shown search results are displayed as markers on Google Maps. Both approaches are highly optimized for collections of spatial knowledge. Therefore, we demonstrate them by means of our UbisEarth ontology.

2 Visualization of Spatial Knowledge and Search Results

Nowadays, more and more spatial knowledge and location-aware data become publicly available due to user-generated content in the Web 2.0. Consequently, improved techniques for identification, visualization and interaction are needed for user interfaces.

2.1 UbisEarth: Visualization of Named Graphical Entities and their Spatial Relations

In the UbisEarth ontology tree (our largest ontology with about 28 million nodes) we display the different instances of the world beginning on top and going down over continents, regions and cities up to even buildings and rooms. In addition, the tree contains many lakes, streams and mountains. In order to add further meaning to the different nodes and to give the tree more structure, we created different role (or relation) nodes as depicted in Figure 1.
The role "‘Country"’, for instance, not only tells us that Europe has 52 countries, but also the other way around, i.e. that Austria, for example, is a country located in Europe. This figure shows another strength of our approach: using a newly defined mechanism to define multilingual labels, which we call UbisLabel, we are able to display not only the English translation of a country's or city's name, but also the appropriate translations into all the languages spoken in the corresponding country.
Furthermore, this tree emphasizes the biggest strength of our visualization technique by pre-processing and displaying limitation nodes, which subsume large numbers of nodes depending on the number of nodes on one tree layer in ten

thousand or hundred packets, respectively. These nodes add additional structure to the tree hierarchy and facilitate an easy-to-use tree browsing. This is one of the big advantages of our approach compared with related work like the GeoTree mentioned above.

Figure 1: Part of the UbisEarth ontology tree

The properties of different locations are also integrated into the tree. For countries we show information like area, capital, currency, spoken languages and population (Figure 2 left), whereas cities can have properties like altitude, latitude, longitude, population, zip codes or dialing codes (Figure 2 right).

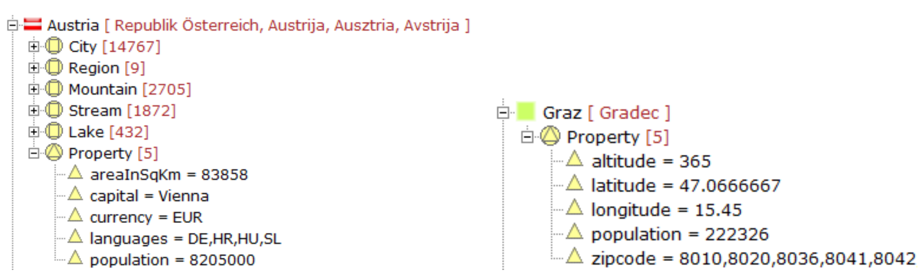


Figure 2: Left: Property nodes for Austria; Right: Property nodes for Graz

Based on this visualization technique, we implemented an extension of the ontology tree representation with additional editing functionalities. This online ontology editor (called UbisEditor) [Loskyll and Heckmann 2009] shall make an end-user friendly collective knowledge building possible. The editor already supports the most important functionalities for editing an ontology: creating new classes or instances, renaming and deleting objects. By using a context menu, we provide an efficient and easy-to-use way of performing these editor actions (Figure 3). With the help of UbisEditor our users could add new translations of location names, fix mistakes or add an instance of their residence, for example.

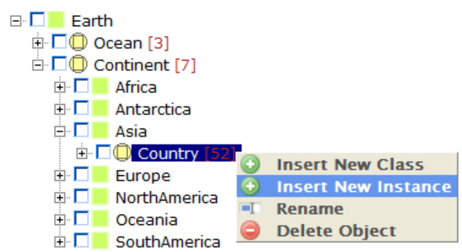
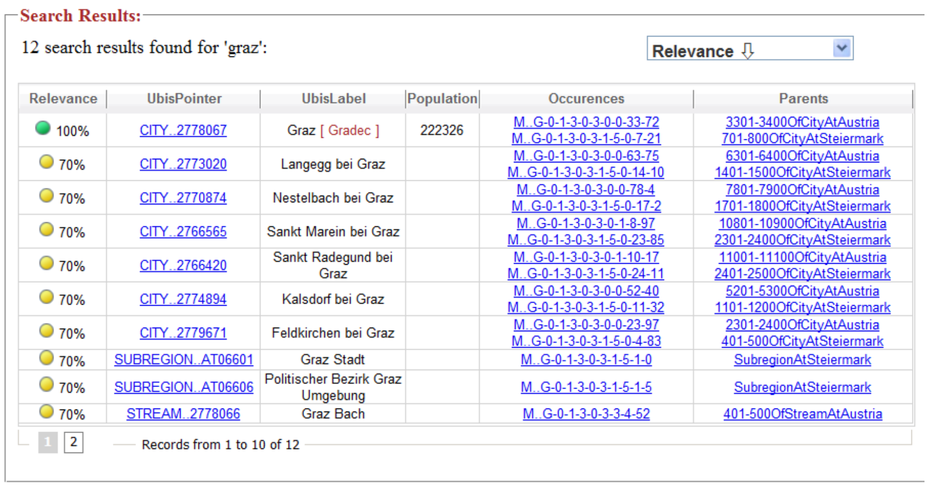


Figure 3: UbisEditor after right mouse button click

2.2 UbisSearch: Adaptive Search Results of Spatial Data

UbisSearch is our search engine, which can be used to find classes, instances and properties of our ontologies. This is an important feature, especially when a user plans to insert a new element into the UbisWorld ontology, but is not sure whether such a concept already exists. We built our own inverted index, in which we assign each term that occurs in a UbisLabel of our ontologies to the corresponding list of UbisPointers (our unique object identifiers). By doing so, we are able to perform search queries in an efficient way. However, the big difference compared with the Google search engine, for example, is that we search over a set of labels instead of documents.

We display the results by using a JavaScript grid, which supports on-demand-loading by using Ajax technologies and paging, i.e. we can present very large search result sets. The grid can be sorted either by relevance, UbisPointer, UbisLabel or population. By default, the information is sorted firstly by relevance and secondly by population (or UbisLabel in case of non-spatial data). Figure 4 shows the search results for the key word “Graz”. In addition, you can easily make sure that this is the Austrian city by looking at the parent column, which shows the different occurrences of this city in our ontology. One of the parents tells us that this is a city in Austria. The idea behind the occurrence column is that we can have multiple occurrences of one location (e.g. a city can appear as located in a country, but also in a region). Selecting one of the occurrence links opens the corresponding node in the ontology tree.



Search Results:

12 search results found for 'graz':

Relevance ⇩

Relevance	UbisPointer	UbisLabel	Population	Occurences	Parents
100%	CITY..2778067	Graz [Gradec]	222326	M..G-0-1-3-0-3-0-0-33-72 M..G-0-1-3-0-3-1-5-0-7-21	3301-3400OfCityAtAustria 701-800OfCityAtSteiermark
70%	CITY..2773020	Langegg bei Graz		M..G-0-1-3-0-3-0-0-63-75 M..G-0-1-3-0-3-1-5-0-14-10	6301-6400OfCityAtAustria 1401-1500OfCityAtSteiermark
70%	CITY..2770874	Nestelbach bei Graz		M..G-0-1-3-0-3-0-0-78-4 M..G-0-1-3-0-3-1-5-0-17-2	7801-7900OfCityAtAustria 1701-1800OfCityAtSteiermark
70%	CITY..2766565	Sankt Marein bei Graz		M..G-0-1-3-0-3-0-1-8-97 M..G-0-1-3-0-3-1-5-0-23-85	10801-10900OfCityAtAustria 2301-2400OfCityAtSteiermark
70%	CITY..2766420	Sankt Radegund bei Graz		M..G-0-1-3-0-3-0-1-10-17 M..G-0-1-3-0-3-1-5-0-24-11	11001-11100OfCityAtAustria 2401-2500OfCityAtSteiermark
70%	CITY..2774894	Kalsdorf bei Graz		M..G-0-1-3-0-3-0-0-52-40 M..G-0-1-3-0-3-1-5-0-11-32	5201-5300OfCityAtAustria 1101-1200OfCityAtSteiermark
70%	CITY..2779671	Feldkirchen bei Graz		M..G-0-1-3-0-3-0-0-23-97 M..G-0-1-3-0-3-1-5-0-4-83	2301-2400OfCityAtAustria 401-500OfCityAtSteiermark
70%	SUBREGION..AT06601	Graz Stadt		M..G-0-1-3-0-3-1-5-1-0	SubregionAtSteiermark
70%	SUBREGION..AT06606	Politischer Bezirk Graz Umgebung		M..G-0-1-3-0-3-1-5-1-5	SubregionAtSteiermark
70%	STREAM..2778066	Graz Bach		M..G-0-1-3-0-3-3-4-52	401-500OfStreamAtAustria

1 2 Records from 1 to 10 of 12

Figure 4: Adaptable search results for "Graz"

In addition to the grid visualization, we display the ten search results that are currently shown on the selected page of the grid as numbered markers on Google Maps (Figure 5). The numbering of the markers make it easy to identify which search result corresponds to which location on the map. We also provide a functionality to show all search results at once as Google Maps markers.

3 Conclusions and Future Work

We have addressed the questions how to represent huge knowledge sets as scalable trees on the web and how the represent particularly spatial search results in an easy-to-use manner. As partial answers, we have presented improved techniques for visualizing huge sets of spatial knowledge with the help of newly defined ontology trees by means of our UbisEarth ontology. Our ontology tree representation approach includes concepts to define relations, properties and structuring nodes in an Ajax-based tree visualization. Furthermore, we described how our adaptable search result grid can be used to efficiently display, browse

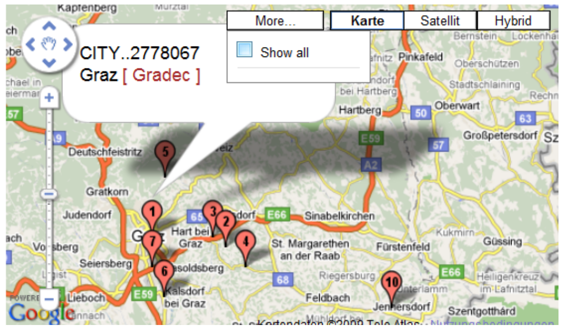


Figure 5: Displaying search results with Google Maps

and sort large sets of search results. These technologies can be tested as registered user at www.ubisworld.org.

In the near future, we are going to perform an extensive evaluation of the ontology tree and search result grid visualization techniques.

References

[Garrett 2005] Garrett, J. J.: "Ajax: A New Approach to Web Applications"; (2005).

[Heckmann 2006] Heckmann, D.: "Ubiquitous User Modeling"; Berlin: Akademische Verlagsgesellschaft Aka GmbH, (2006).

[Heckman et al. 2007] Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., Kröner, A. : "The User Model and Context Ontology GUMO Revisited for Future Web 2.0 Extensions"; In: CEUR Workshop Proceedings (2007).

[Loskyll 2007] Loskyll, M.: "Ontological and Ajax-based Extension of UbisWorld"; Bachelor thesis, Chair for Artificial Intelligence, Prof. Dr. Dr. h.c. mult. Wahlster, Saarland University, (2007).

[Loskyll and Heckmann 2009] Loskyll, M., Heckmann, D.: "UbisEditor 3.0: Collaborative Ontology Development on the Web"; In: Proceedings of the Hypertext 2009 workshop on Web 3.0 - Merging Semantic Web and Social Web, Torino, Italy (2009).

[O'Reilly 2005] O'Reilly, T.: "What is Web 2.0"; (2005).

[Pease 2002] Pease, A., Niles, I., Li, J.: "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications"; Workshop on Ontologies and the Semantic Web, (2002).

[Wahlster and Dengel 2006] Wahlster, W. and Dengel, A.: "Web 3.0: Convergence of Web 2.0 and the Semantic Web"; Telekom Technology Radar II, Juni, (2006).